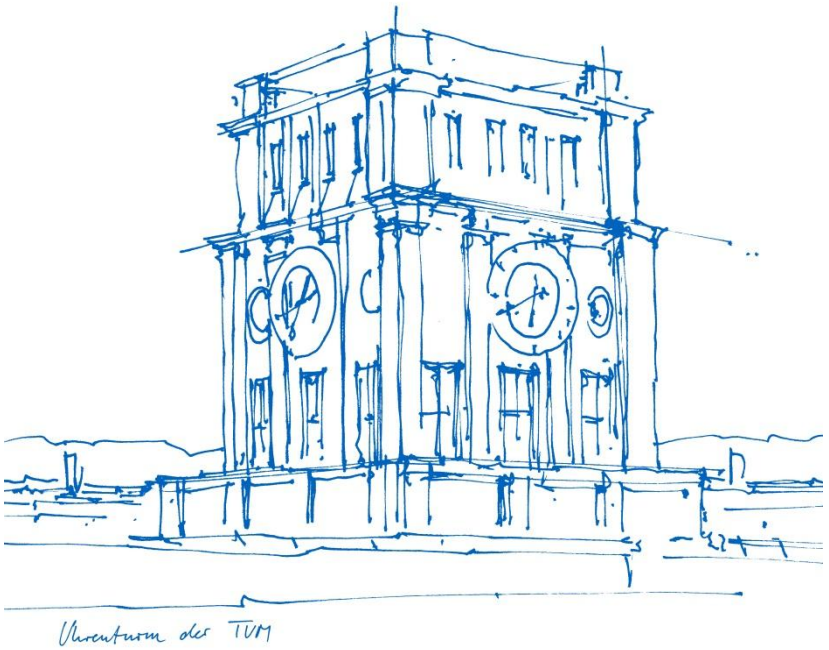
Multi-Sensory Based Robot Dynamic Manipulation: Final project

Technical University of Munich

Chair for Cognitive Systems

Prof. Dr. Gordon Cheng



Student Name:
Gintautas Palinauskas

Advisor:
M.Sc Simon Armleder

Control Decisions **2**

Possible improvements **3**

Appendix **6**

Update Flowchart 6

Joint Control Flowchart 7

Modified operational control 8

Calculating Tracking Orientation 9

Hyperparameters 9

Control Decisions

Summary:
Orientation representation: quaternions.
First positioning: joint control.
consider Qd, Qdp, Qdpp from polynomial interpolation (PI).
Second positioning: operational control with impedance.
Circular motion: consider Xd from the trajectory generator.
Repositioning: consider Xd, Xdp linear part, Xdpp linear part from PI and slerp.
Calculate tracking orientation using Euler angles.

The decision to use quaternions was based on the fact that they represent rotations uniquely, if we exclude -1 symmetry, as in comparison with Euler angles, which does not have a unique representation of orientation and also adds additional singularities to the system. The task is then divided in two motions. First, using joint contro leaving the singularity position in a predefined position. During the control the robot takes into account joint angles, velocities, accelerations goals from the polynomial path generator. Second motion part is a combination of operational control for following the circular trajectory and impedance control for avoiding obstacles. For the following path only pose goals are considered for simplicity. Moreover, the repositioning of the robot end-effector, when pushed away by the obstacles, takes into account linear positions, velocities, accelerations as well as angular orientations, generated by Eigen method slerp and polynomial interpolation. The flowchart is given in the appendix.

To track the targets only the orientation of the robot end-effector needs to be calculated and used as a goal orientation in the control loop. The angles are derived by considering the fig 1. and calculating the angles alpha and beta using basic trigonometry. The angles correspond to rotation about x axis and y axis respectively. The flowchart is given in the appendix.

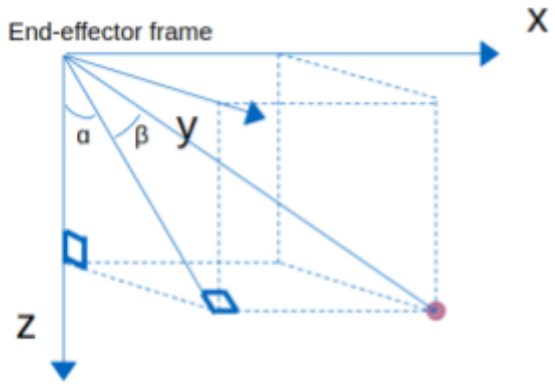


Figure 1. Euler angle calculation diagram for tracking.

Possible improvements

In the current state the system is sensitive to parameter changes, so the following changes in the future could make the system more stable. The first improvement would be to add impedance control with respect to joint 3, because joints 3 and 6 are the collision points for the obstacles and impedance control with respect to joint 6 is already added. Second change would be to divide the ros package into smaller subpackages for different controls and visualization. Moreover, after analyzing the plots given in the appendix below, a low-pass filter would smoothen the control. Furthermore, after some test runs the system fails to switch from operational control to impedance control during repositioning after avoiding the obstacles. Lastly, tuning the hyperparameter given in the appendix would further increase the robot's stability and reliability.

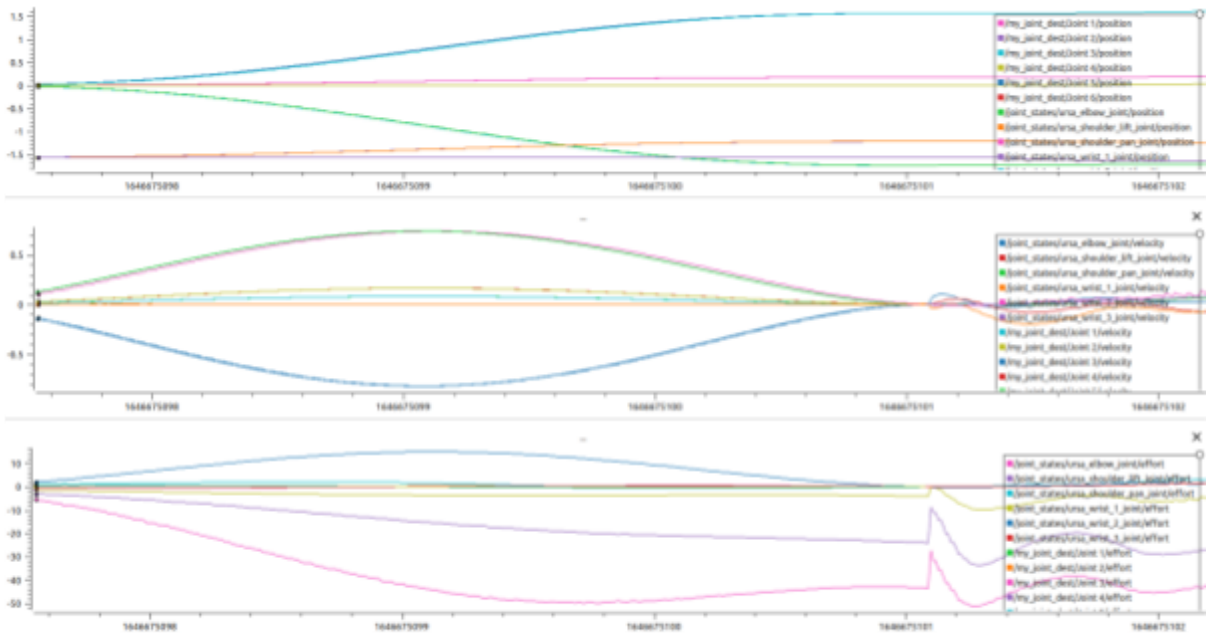


Figure 2. Joint control positioning plots

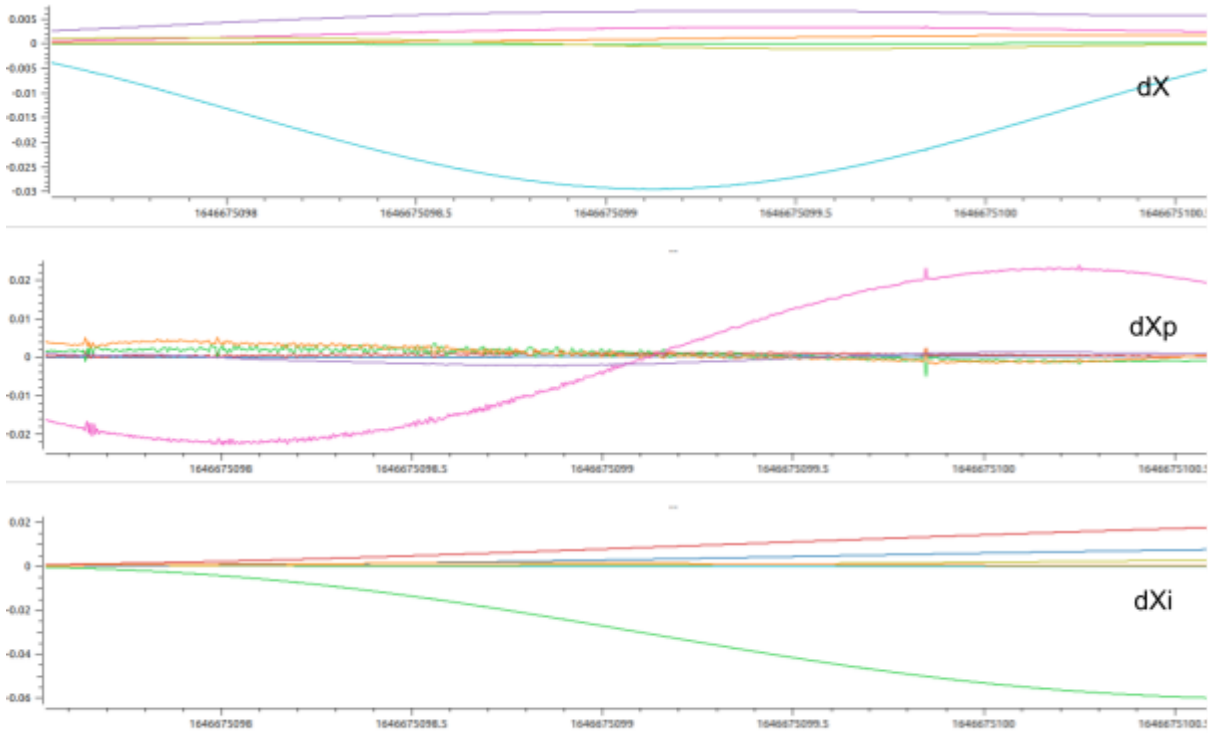


Figure 3. Joint control error plots.

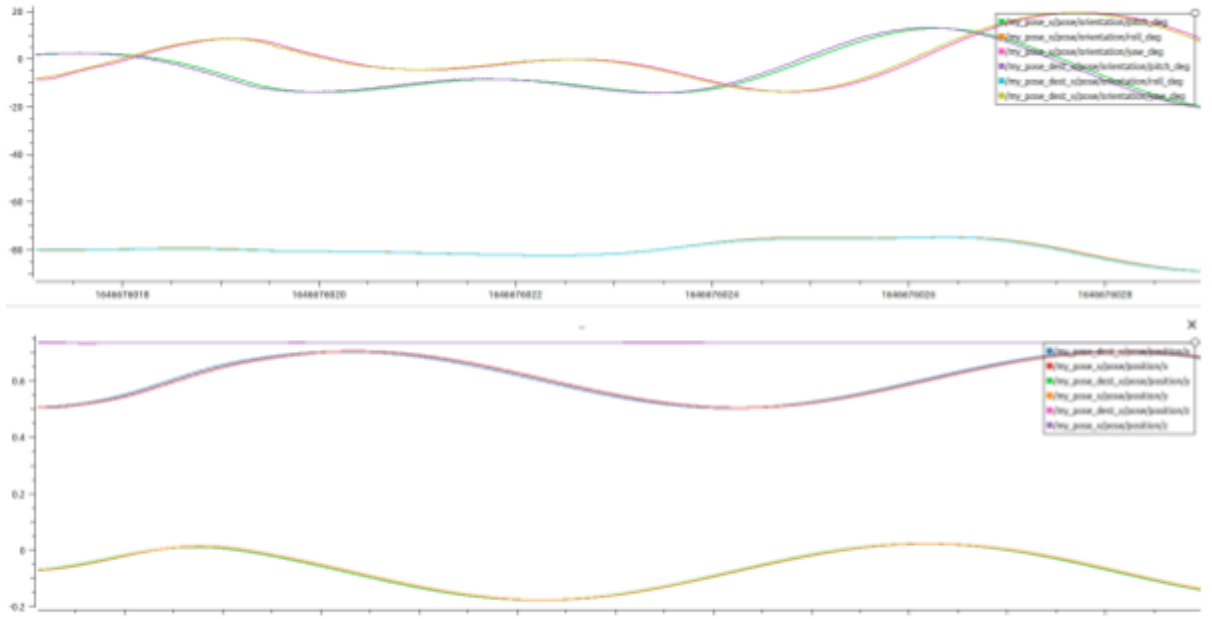


Figure 4. Operational control pose plots.

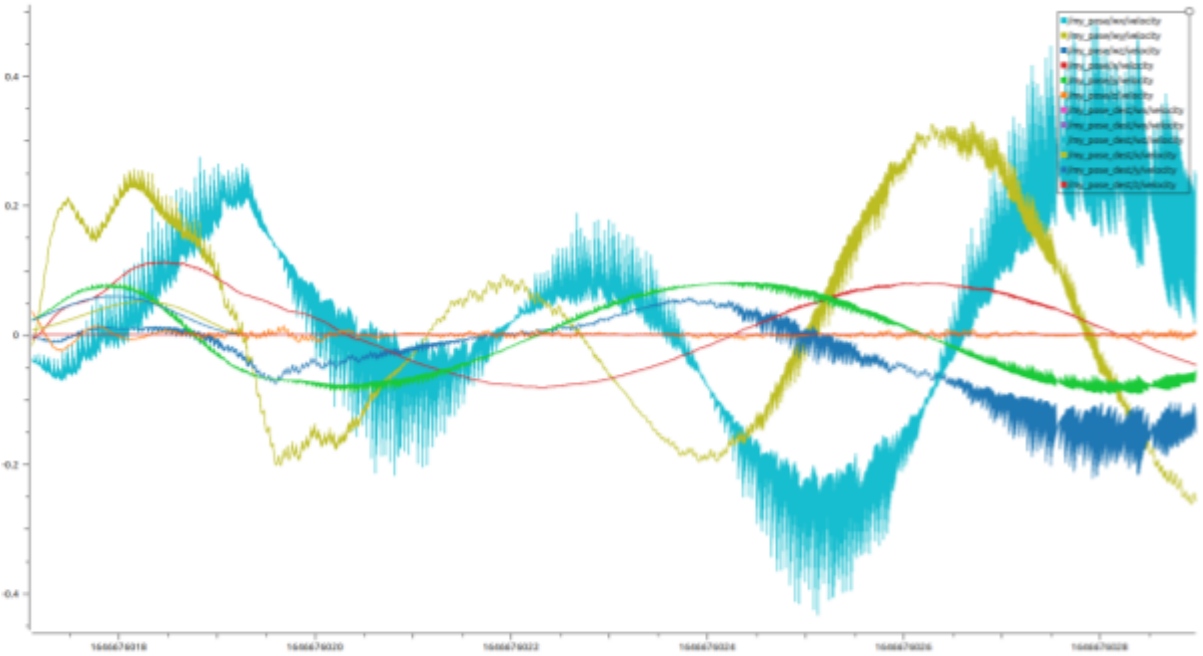


Figure 5. Operational control pose velocity plots.

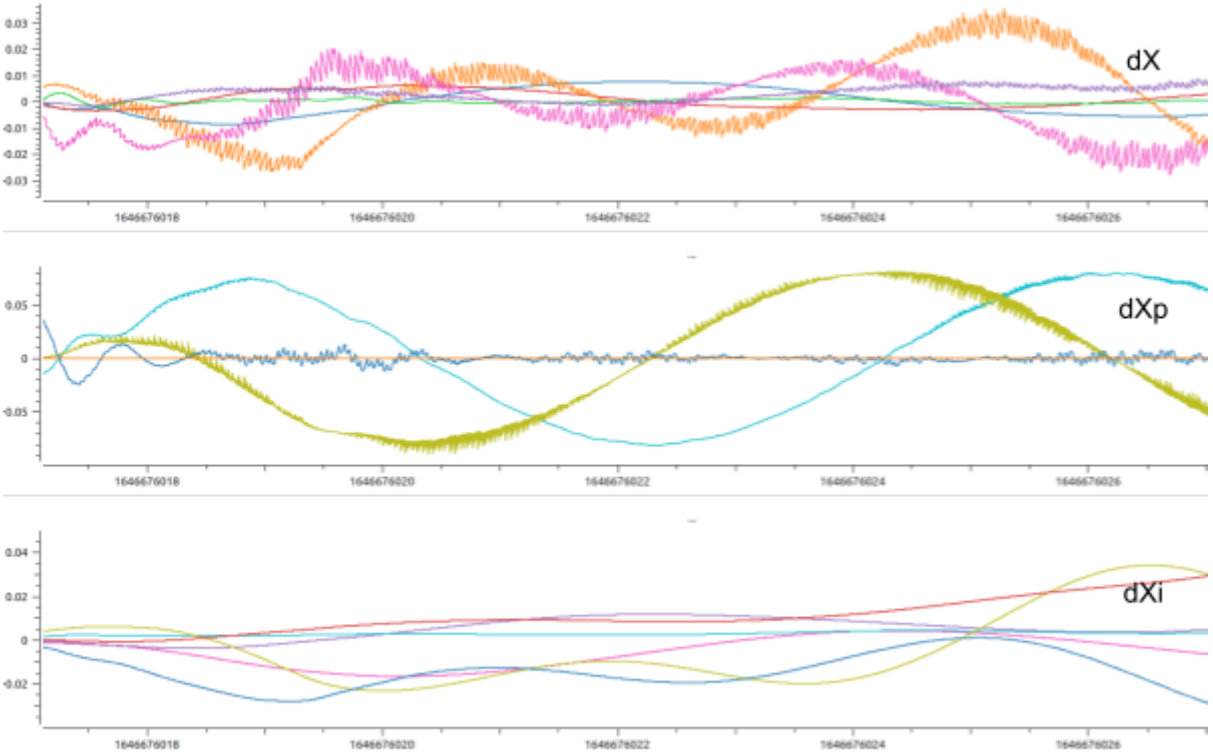
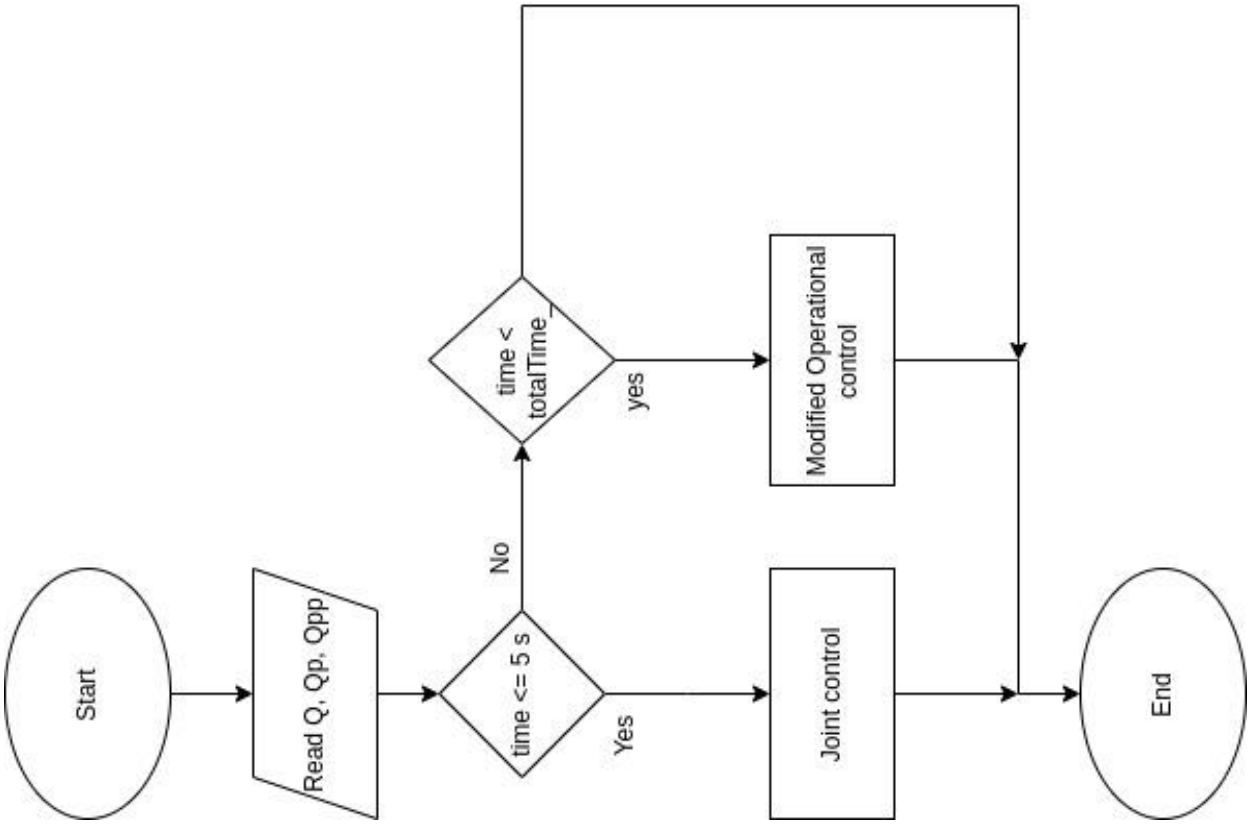


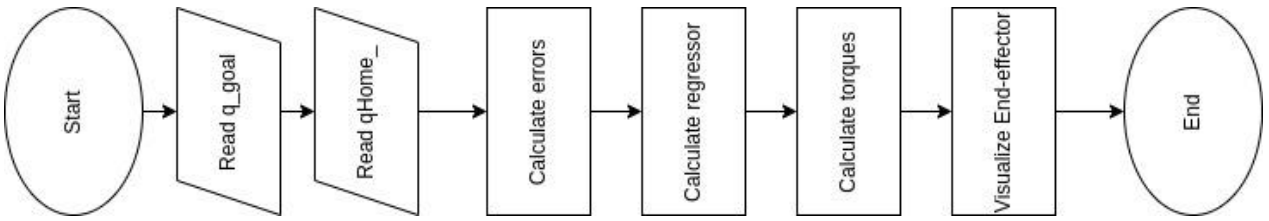
Figure 6. Operational control error plots.

Appendix

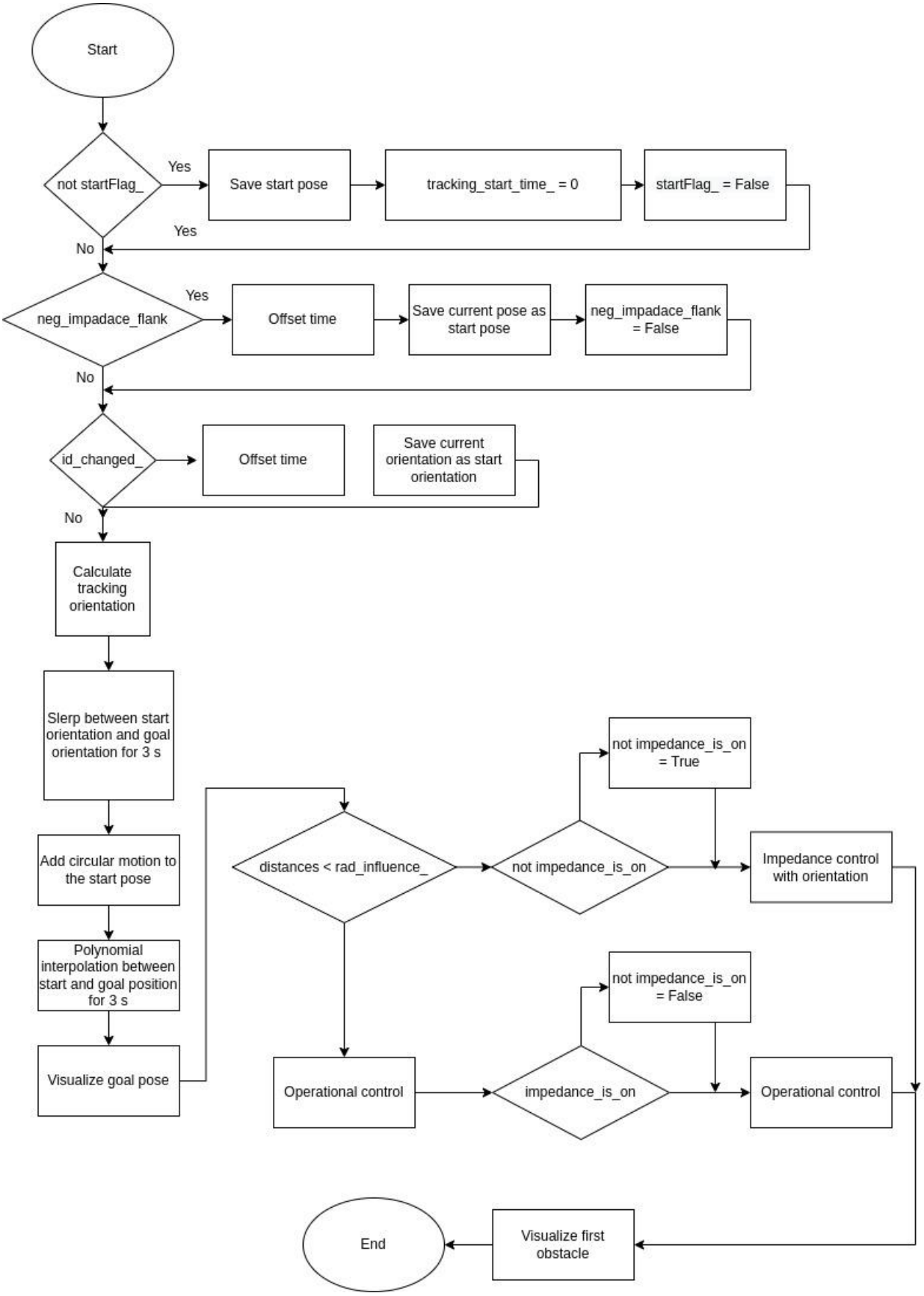
Update Flowchart



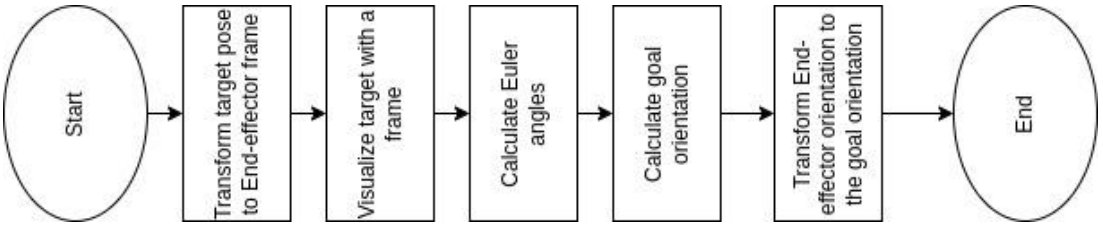
Joint Control Flowchart



Modified operational control



Calculating Tracking Orientation



Hyperparameters

Joint control:
p: [300,300,300,100,100,100]
d: [30,30,30,5,5,5]
i: [1,1,1,1,1,1]
gamma: 0.001

Operational control:
p: [20,20,20,15,15,15]
d: [40,40,40,25,25,25]
i: [0.01,0.01,0.01,0.01,0.01,0.01]
gamma: 1

Impedance control linear:
p: [100,100,100]
d: [300,300,300]

Impedance control angular:
p: [50,50,50]
d: [20,20,20]

influence radius: 0.3
multiplier factor: 20
repelling force function: 1/x

time step: 0.002